DISKMATE.

INSTRUCTIONS. Never connect or disconnect a cartridge while the power
is on.

The Diskmate cartridge features a reset button and a small switch on
the rear of the unit. The switch effectively makes Diskmate two
cartridges in one so these instructions will treat it as such.

Switch Down. In this position you will be presented with a menu on
power up as follows;

F.   Fast Format. This option will format a disk in about 10 seconds.
     Simply select 'F' and then enter disk name and ID when prompted.
     Note. Formatting a disk will destroy any previous data on it.

I.   Install fastload. This feature will load programs at about 5 times
     normal speed. When you press 'I' a dot will appear alongside to
     indicate that the module installed. If you now 'Q'uit to basic you
     can use the feature.
     NB Fast load will not speed up some software particularly programs
     that already feature their own fast loader.

U.   Utility commands. Select 'U' and a dot will appear to indicate
     that the module is installed. If you now 'Q'uit to basic you can
     use the following keywords;

$	Directory.
@$	Directory.
DIR	Directory.
@SO:filename	Scratch file.
@RO:newname=oldname	Rename a file
@IO	Initialize disk drive.
@VO	Validate a diskette.
@	Read error channel
/filename	Load basic program status key.

C.   4 minute copier. Copy a complete disk even some protected
     software. Simply follow the on screen prompts.

Q.   Quit will take you back to basic. Can be re entered with reset
     or SYS 32802.

Switch up. In this position on power up or reset you will enter into
the Micromon monitor. The usual proceedure might be to load the
program to be veiwed and then reset to enter Micromon. The first screen
will display the present register status and prompt for a command.
A complete list of Micromon commands are attached.

Notes;

1, All capital letters must be typed as shown. Lowercase letters stand
for information to be supplied by the user. ie (hh= any Hex value)

2, All numbers must be entered in Hex (with no $). Except in number
conversion functions.

3, The brackets ¡¿ around an item mean that it is optional. (Don't
type the brackets.)

Output Commands

P – (PRINT) Switches all output from the screen to the printer (device 4) or vice versa. You must hit return TWICE after the P before entering next command. Use P again to return output to screen. If the last line of printing is 'stuck' in the printer buffer (delayed printing), switching output to printer and back again should cause it to print.

OPEN filenum, dev [,s.a.] [,"string"] :CMD filenum – (General output) If the "P" command fails to work correctly with your printer, exit to BASIC and enter this line. Use the correct device number and s.a. (secondary address, if needed) for your printer. If a string is specified, it will be used as a title. Reenter MICROMON and perform the desired functions. When you're finished with the printer, exit to BASIC and enter **PRINT#filenum :CLOSE filenum**. This technique can also be used to store output in a disk file, by giving the appropriate device, etc. The string will be used as a file name. Be sure to open the file for writing by putting ",W" in file name after the file type (P, S or U).

Debugging Commands

G [addr] – (GO) Restores the values of the registers, etc. that were saved when MICROMON was entered (see R command) and then executes an ML routine. If [addr] is given, execution begins at that location. Otherwise, execution begins at the location given by the saved program counter. The ML routine must end with a BRK instruction in order to return to MICROMON.

W [addr] – (WALK) Restores the saved values of the registers, etc. and begins executing ML code one instruction at a time. Prior to executing an instruction (except the first), the following values are displayed: the status, A, X, and Y registers; the stack pointer; the program counter (address of next instruction); and finally, the hex and mnemonic for the next instruction. If [addr] is specified, execution begins at that location. Otherwise the saved value of the program counter is used. Pressing any key (except J and STOP) will execute the next instruction. Pressing the "J" key will execute a subroutine uninterrupted, until an RTS instruction is encountered. Executing a BRK instruction or hitting the STOP key will return to normal MICROMON command mode and save the current register, etc. values.

Q [addr] – (QUICK TRACE) Exactly like the "W" command except: execution is continuous; the instructions executed are not displayed; use STOP and "=" keys simultaneously to return to MICROMON; the "B" command can be used to set an optional "breakpoint".

B addr [hhhh] – (BREAKPOINT SET) Sets up a "breakpoint" for the "Q" command. When execution reaches the address given, a BRK instruction is executed to return to MICROMON. If [hhhh] is specified, the BRK is executed only after the given address has been executed that number of times.

Exit Commands

X - (SOFT EXIT) Exits MICROMON to BASIC, leaving the BRK and IRQ vectors as set by MICROMON. This allows you to 'hot-start' MICROMON by executing a BRK instruction from BASIC, using SYS 8 for instance. The "X" command will leave PROMOS enabled.

E - (HARD EXIT) Exits MICROMON to BASIC but restores the previous contents of the BRK and IRQ vectors (set by PROMOS). MICROMON must be cold-started to reenter it. This command will leave PROMOS enabled.

K - (KILL) Exits MICROMON to BASIC and restores the power-up values of the BRK and IRQ vectors. Same effect as RUN/STOP-RESTORE.

G FCE2 - (RESET) Totally resets the computer.

File commands

L [alt. addr] "file" [,dev] - (LOAD) Loads a file from disk (no tape). If [,dev] is omitted, device 8 will be selected. If [alt. addr.] is specified, the file will be loaded at that address rather than its normal load address. This is very useful for EPROM work. After loading, the ending address of the program (+1) will be printed.

shift-L [alt. addr] "file" [,dev] - (Dummy LOAD) Same as L, but does not actually load the file. Used to find the ending address of a program (+1).

V "file" [,dev] - (VIEW) Prints the load address of the file, but does not load it.

S "file", start, end+1 [,dev] - (SAVE) Saves an area of memory to disk. Note that the end address you give must be 1 greater than the actual end.

Memory Commands

A addr instruction - (ASSEMBLE) Assembles one ML instruction into memory starting at the specified address. All addresses in the INSTRUCTION must be preceded by a $. Also automatically inserts an "A" on the following line to prepare for assembling another statement. Hit return or move to a blank line to end assembly.

N code-start code-end offset ref-begin ref-end [W] - (NEW LOCATION) Used to relocate a section of ML code - sort of like a RENUMBER command in BASIC. After moving the code with the "T" command, this command will adjust absolute memory references (not branch locations) to reflect the code's new location. The area specified by code-start and code-end will be searched for any (three-byte) instructions which reference the area specified by ref-start and ref-end. Any references found will be adjusted by adding the offset value. To calculate the proper offset, subtract the original start address from the new start address, using the "-" command. The [W] option specifies that a table of two-byte vectors (Words) is being searched rather than ML code. See the chapter on reusing the tape routines in the KERNAL for an example of using this command.

O instr-addr target-addr - (OFFSET) Used to calculate the offset for branch instructions. Instr-addr is the address of the branch instruction itself and target-addr where you want it to branch to. This command is pretty useless since MICROMON automatically calculates the offset for you when you assemble a branch instruction. Don't use this command for calculating offsets for the "N" command.

$ hhhh - (HEX CONVERSION) Converts the hex number hhhh into decimal, ASCII characters and binary.

 # ddddd - (DECIMAL CONVERSION) Converts the decimal number ddddd into hex, ASCII characters and binary.

% bbbbbbbb - (BINARY CONVERSION) Converts the binary number bbbbbbbb into hex, decimal, and ASCII characters.

" a - (ASCII CONVERSION) Converts the ASCII character a into hex, decimal and binary.

+ hhhh hhhh - (HEX ADDITION) Adds the two hex numbers and gives a one-byte hex result. If the result is over $FFFF, only the low byte is given.

- hhhh hhhh - (HEX SUBTRACTION) Subtracts the second hex number from the first.

& start end - (CHECKSUM) Checksums an area of memory by ADDing together all of the bytes. The result is limited to two hex bytes.